Biotic Prediction

Building the Computational Technology Infrastructure for Public Health and Environmental Forecasting

Concept of Operations

BP-CONOP-1.9

Task Agreement: GSFC-CT-1

December 4, 2002

Contents

List of Tables

List of Figures

1 Overview

1.1 Introduction

This project will develop the high-performance, computational technology infrastructure needed to analyze the past, present, and future geospatial distributions of living components of Earth environments. This involves moving a suite of key predictive, geostatistical biological models into a scalable, cost-effective cluster computing framework; collecting and integrating diverse Earth observational datasets for input into these models; and deploying this functionality as a Web-based service. The resulting infrastructure will be used in the ecological analysis and prediction of exotic species invasions. This new capability will be deployed at the USGS Midcontinent Ecological Science Center and extended to other scientific communities through the USGS National Biological Information Infrastructure program.

1.2 Referenced Documents

Table 1. Referenced Documents

Document Title	**Version**	**Date**
Software Engineering / Development Plan	1.2	2002-09-26
Software Requirements Document	1.2	2002-10-17

1.3 Document Overview

This document, the *Concept of Operations*, describes the *Invasive Species Forecasting System* (ISFS) from an operational point of view. It is not intended to imply a design or convey implementation requirements for the functionality described herein.

It provides a brief introduction to the functional architecture of the system through its intended external interfaces, and describes a high level description of the elements that will be combined in the prototype. It also presents the physical architecture of the system and discusses the operation of the system through those physical components.

Finally it describes the system's operational scenarios through a series of "use cases." These use cases explain in detail how we think users will interact with the system, and the types of system functions they will initiate through their tasks. The use cases are illustrated with early renditions of the user interface elements that will support their operation.

2 ISFS Functional Architecture

2.1 System Description

The ISFS system will have users from government agencies, universities, industry, and the general public. To the users, the ISFS is deployed as a web browser based system that will present options for applying a series of models to available datasets yielding predictive result sets. The system can ingest data from different sources and in different formats and existing models can be either run, or new ones created. The system outputs maps and additional information depicting the applied model and the predicted species distributions.

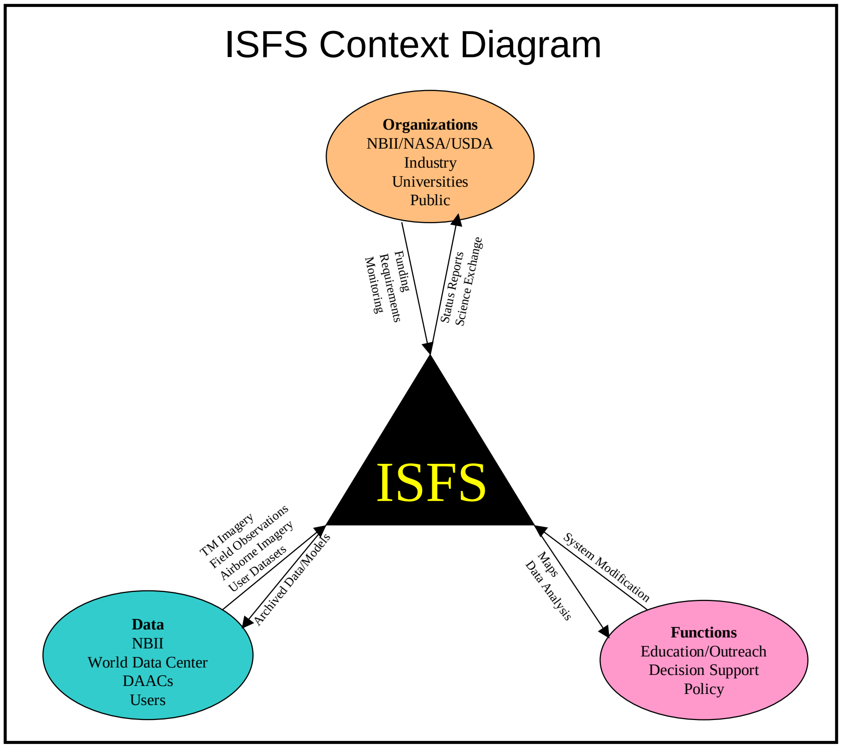


Figure 1. Context Diagram

2.2 External Interfaces

The primary means of interface to the ISFS will be a W3 standards compliant web browser. All GUI development efforts for external interfaces will be through web-based client architecture that uses HTTP as the primary means for interacting with the system. Supplmentary interfaces for ingest may include a secured FTP push/pull technique that will be scheduled through the GUI.

Standing orders for required data products will be accomplished through subscriptions to the varied data sources including the DAAC, NBII, USGS, and other data sources TBD. Different levels of access for public, model user, model builder, and developer will be established and documented in the SRD.

The user interface is implemented through an HTTP connection to browser-based GUI to select models and datasets, and to apply the selected model to the selected dataset. The GUI is also the vector for returning processed output to the end user.

2.3 Internal Architecture

The overall internal architecture of the Invasive Species Forecasting System is presented in Figure 2. It consists of three major conceptual layers: (1) a Front End Layer that provides a suite of graphical user interfaces to the various components of the system that must be accessed by the various classes of users of the system, (2) an Application Layer whose subsystems support the activities, computations, and workflows that define the ISFS modeling process, and (3) a Backend Layer that provides persistent archive storage for both system and user needs. Each of these major layers and their subsystems are explained in greater detail in the sections that follow.

Figure 3 depicts the flow through the system.

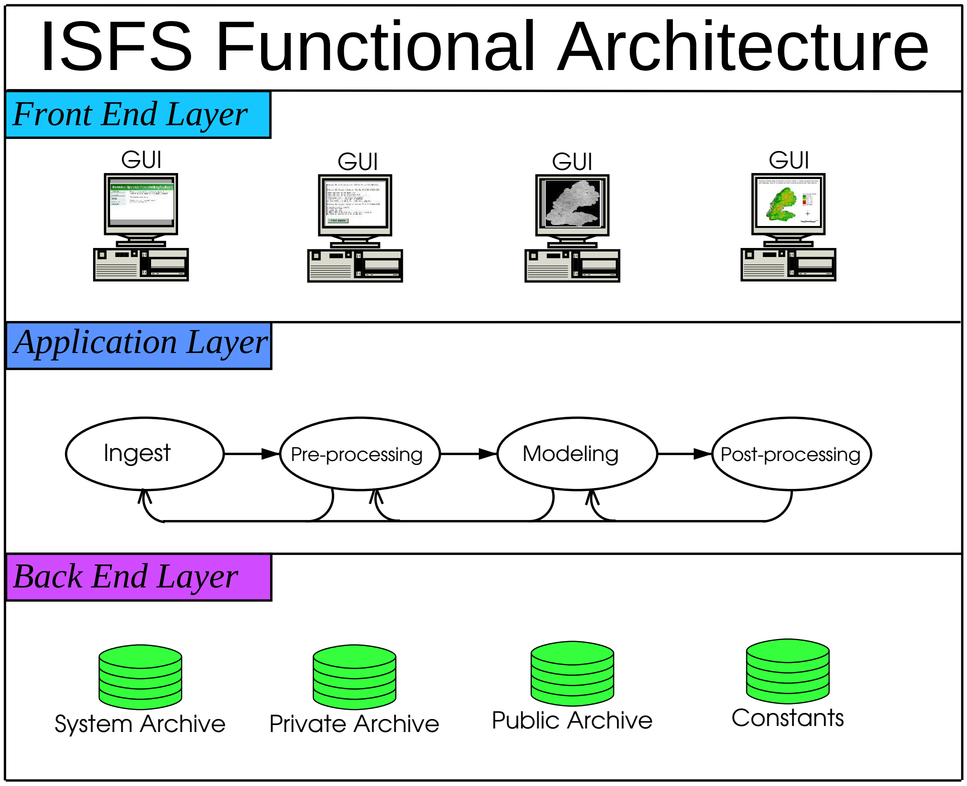


Figure 2. ISFS Functional Architecture

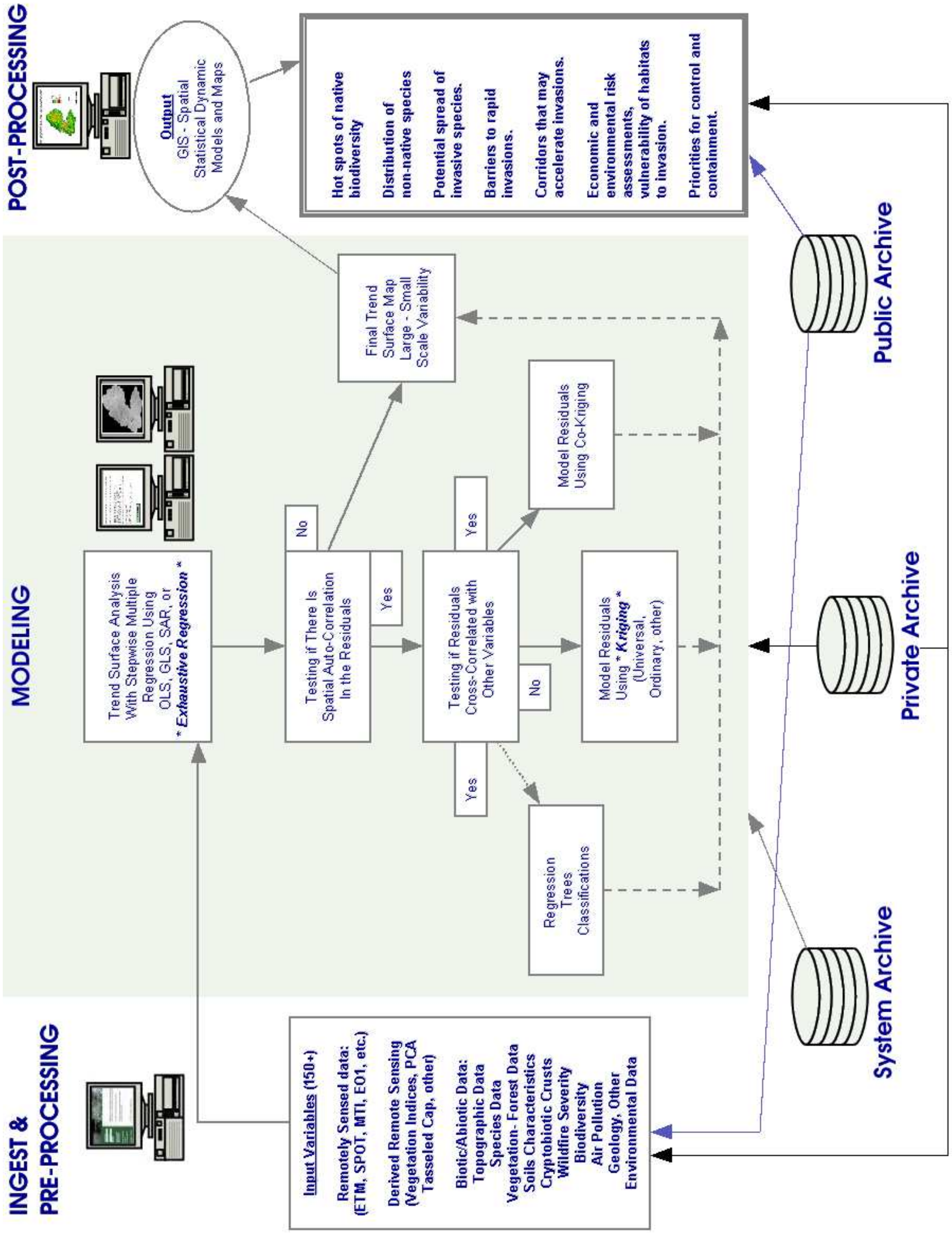

INGEST &
PRE-PROCESSING
Input Variables (150+)
Remotely Sensed data: (ETM, SPOT, MTI, E01, etc.)
Derived Remote Sensing (Vegetation Indices, PCA Tasseled Cap, other)
Biotic/Abiotic Data:
Topographic Data
Species Data
Vegetation- Forest Data
Soils Characteristics
Cryptobiotic Crusts
Wildfire Severity
Biodiversity
Air Pollution
Geology, Other
Environmental Data
MODELING
Trend Surface Analysis With Stepwise Multiple Regression Using OLS, GLS, SAR, or * Exhaustive Regression *
Testing if There Is Spatial Auto-Correlation In the Residuals
No
Yes
Testing if Residuals Cross-Correlated with Other Variables
Yes
No
Yes
Model Residuals Using Co-Kriging
Model Residuals Using * Kriging * (Universal, Ordinary, other)
Regression Trees Classifications
Final Trend Surface Map Large - Small Scale Variability
POST-PROCESSING
Output
GIS - Spatial Statistical Dynamic Models and Maps
Hot spots of native biodiversity
Distribution of non-native species
Potential spread of invasive species.
Barriers to rapid invasions.
Corridors that may accelerate invasions.
Economic and environmental risk assessments, vulnerability of habitats to invasion.
Priorities for control and containment.
System Archive
Private Archive
Public Archive

Figure 3. ISFS Data Flow

2.3.1 Front End Layer

The ISFS front end will support a variety of interfaces that allow controlled and tailored access to the various subsystems of the overall system. The front end consists of a graphical user interface subsystem with both client- and server-side components.

2.3.1.1 User Interface Subsystem

The User Interface Subsystem provides a way of managing information and performing analyses by means of dynamically constructed activity-and-information spaces, or role-based views. The various role-based views are delivered through dynamically constructed, personalized Web pages.

A profile database maintains a profile of each user's status and preferences. Activities are implemented by a library of routines accessed through the controls on a web based forms interface. Each user is assigned a particular role, which enables them to perform certain actions but not others, and to view certain types of information but not others. Roles include the following:

- "Administrator" — This role provides complete access to the ISFS web system. An administrator would be able to manage other users and have access to their workspace, preferences and authorization information.
- "Model Builder" — This role is intended for the user that wishes to configure and tailor the model related tasks. The Model Builder is someone who has been authenticated and authorized to upload and ingest data in preparation for other users to perform repeated model runs. The Model Builder has registered with the system and maintains an active logon name and password to access the system. The Model Builder maintains a profile within the system that remembers the Builder's data selections and makes those selections available to the Builder upon login.
- "Model User" — This role is intended for a user to select specific input data and run it through a previously configured ISFS model run. The Model User has registered with the system and maintains an active logon name and password to access the system. The Model User maintains a profile within the system that remembers the User's data selections and makes those selections available to the User upon login.

The roles will be maintained by incorporating the Lightweight Directory Access Protocol (LDAP). LDAP is a protocol for accessing online directory services over the TCP/IP network protocol, and can be used to access standalone LDAP directory services or directory services supporting the X.500 standard. It provides a standard way for Internet clients, applications and Web servers to access directory listings of thousands of Internet users (http://wp.netscape.com/newsref/pr/newsrelease126.html).

2.3.2 Application Layer

The application layer consists of four subsystems that support the activities, computations, and workflows of the ISFS: Ingest, Preprocessing, Modeling, and Postprocessing. These subsystems may be invoked by the user sequentially, arbitrarily, and iteratively to support the various steps that make up the ISFS modeling process. They may also be invoked automatically by using processing scripts as explained below.

2.3.2.1 Ingest Subsystem

The ingest subsystem will serve as the initial “entry point” for all data used in the system. The data fall into the three main categories: field point measurements, imagery, and ancillary layers. These categories will be further defined in the Requirements Analysis Document. Common to the three categories is that all data ingested into the system will be associated with some geographic location. There will be a validation step to verify the integrity of the data before ingest, and every effort will be made to ascertain that the data originated from an authoritative source.

Within the subsystem users will be able to upload field data in a tabular form using standard templates provided by the Invasive Species Forecasting System. These templates will ensure that all system required fields will be captured and will be in an accessible format (such as a spreadsheet, database, or simple ASCII list). Satellite data will be included primarily from external satellite data archives but also user-supplied satellite data or airborne imagery may be used. The primary source for ancillary layers will initially be USGS but here too the ingest system will allow user-supplied ancillary layers to be incorporated into the system.

Data ingested into the system fall into three categories:

1. A Tabular file (typically from field observations and containing latitude, longitude, date information, and observed values at the given point)
2. Raster layers (typically image data or GIS “grids” which extend over a large area and have one or more layers of information and can have different spatial resolution for the pixel size).
3. A boundary area specifying the region of interest (such as boundary of a national park or monument). The area can be defined by a vector GIS file, a binary raster image, or simply a list of bounding coordinates.

The system will allow only one tabular file and one boundary area but multiple raster data files for each modeling scenario. The format for the tabular data will be a comma delimited ASCII file containing the following elements for each ground observation (hard returns are given here for clarity in the document, the actual values would be on one line):

- siteID,
- upper left X, upper left Y,
- upper right X, upper right Y,
- lower left x, lower left y,
- lower right x, lower right y,
- large plot ID (if the site is part of a larger plot, if not fill with something like -999)
- date of collection,
- observed variable 1, ..., observed variable N

While the upper right, upper left, lower right and lower left may be too close to distinguish with the given GPS accuracy (that is, they would all be reported as the same values), this format would allow larger plots to explicitly define the given area (as opposed to a single point) and thus allow more explicit extraction of the image data for an appropriate area.

Mechanisms for data acquisitions will be secured ftp-push for any user supplied data or automated secured ftp pull from the external archives. Users will be issued usernames and will be authenticated through passwords. User activity and preferences will be logged and archived in the system. A specific list of external archives and required data sets will be established and maintained as part of the ingest subsystem. The interfaces to these archives will be negotiated and a thorough understanding of source and target schema will be included in the interface agreement. The ingest subsystem will monitor the number and volume of data brought into the system, with an ability to break this down by location, user, and external archive. Formal procedures for ingest of these data will be documented as the data/source specific requirements become evident. As a general standard operating procedure we will capture metadata and QA type data that will describe each file to be ingested.

For the purpose of establishing the baseline canonical example, we will assume that the datasets exist and are stored locally on our servers. Once the HPC technology has been applied and proven we can expand our ingest routines to include additional themes (ancillary layers) and interfaces with government and university databases. Formal procedures for ingest of these data will be documented as the data/source specific requirements become evident. As a general standard operating procedure we will capture metadata and QA type data that will describe each file to be ingested and write that into the file header or store it in a database record associated with the unique file identifier.

2.3.2.2 Preprocessing Subsystem

The pre-processing stage manipulates the data resulting from the ingest stage into a format/structure that can be used by the modeling component of the system. Limited pre-processing is needed for the tabular data, since the format will be specified in the ingest phase.

The subsystem may perform resampling if the input raster layers are not at the same resolution. The merged data product will be written to the archive in a common analysis format, possibly GeoTIFF.

The primary component of the pre-processing stage will be to extract, for each site (as defined by the coordinates in the tabular file), the information from each raster layer for that site. These values extracted from the raster layers will be appended as columns to the ingested tabular file. The results of this pre-processing step will be referred to as the *"merged tabular data"*.

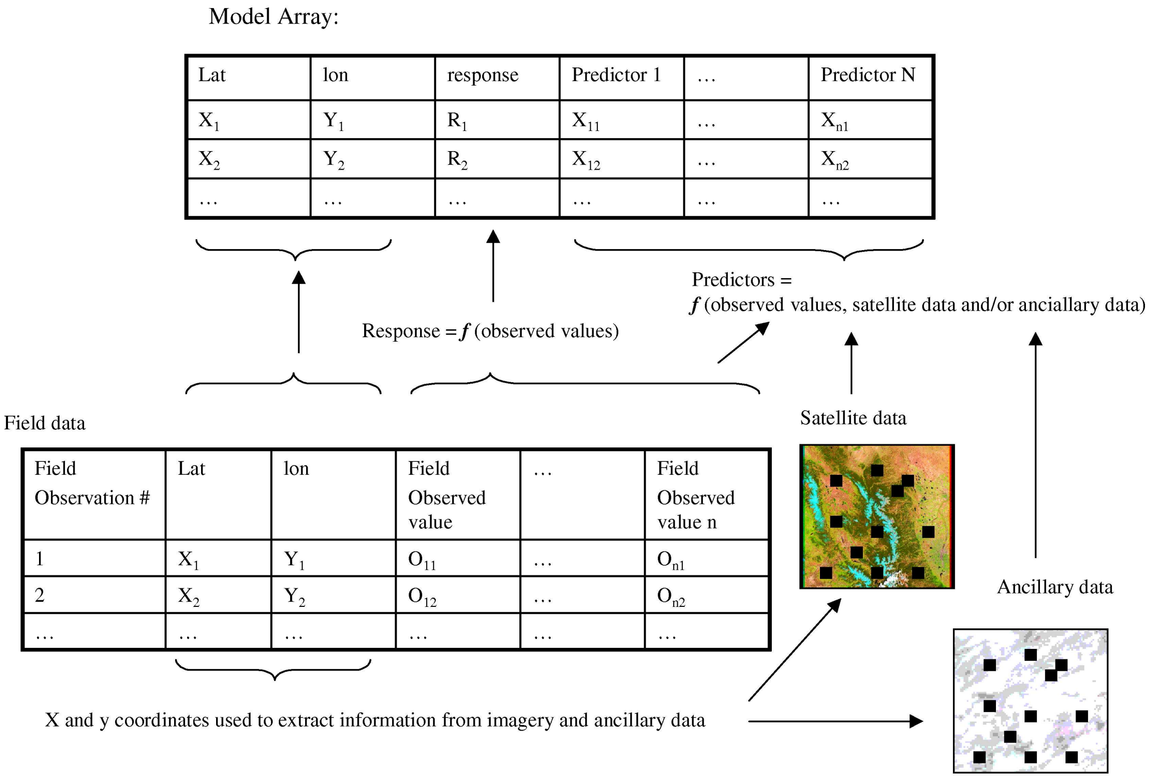
A secondary component of the pre-processing stage will be to create new variables from existing columns of the merged tabular file. These new variables will either be pre-programmed or user-defined functions of the existing columns. The results of this step will be referred to as the *"merged, augmented tabular file"*.

Either a "merged tabular file" or a "merged, augmented tabular file" is required to go on to the modeling stage.

2.3.2.3 Modeling

Modeling in the first version of the ISFS will be empirical in nature and utilize statistical techniques. The general theme of the modeling will be to predict a certain species migration through or invasion of habitat based on remote sensing imagery and ancillary data layers. The modeling subsystem will support five basic workflow activities:

Step 1 — Data Array Construction. Each model will require an array containing the response, or dependent, variable (the "Y" variable) and a set of predictor, or independent, variables (the "X" variables). Constructing the data array needed for modeling will start with a set of geographic coordinates. These will likely come from the geographic coordinates associate with the tabular field data of interest. In addition to the coordinates, the Y variable will be extracted from the tabular field data, either directly or as a function of one or more elements in the field data. The X variables will come from any of the three data sources. X variables from the field data can also be directly extracted or be a function of one or more elements from the data. X variables will be extracted from the satellite and ancillary data by using the coordinate information from the field data to extract values from the imagery for the corresponding pixels. These X variables can come directly from the satellite or ancillary data or be a function of one or more satellite or ancillary variables. A diagram of the modeling array is shown in Figure 4.



Model Array:

Lat	lon	response	Predictor 1	…	Predictor N
X_1	Y_1	R_1	X_{11}	…	X_{n1}
X_2	Y_2	R_2	X_{12}	…	X_{n2}
…	…	…	…	…	…

Field data

Field Observation #	Lat	lon	Field Observed value	…	Field Observed value n
1	X_1	Y_1	O_{11}	…	O_{n1}
2	X_2	Y_2	O_{12}	…	O_{n2}
…	…	…	…	…	…

Predictors =
f (observed values, satellite data and/or anciallary data)
Response = *f* (observed values)
Satellite data
Ancillary data
X and y coordinates used to extract information from imagery and ancillary data

Figure 4. Schematic of Modeling Array

Step 2 — Model Selection and Fitting. Once the array is constructed, model selection will involve screening X variables to see which are most related to the Y variable. Methods to be considered are graphical exploratory analysis, stepwise regression, and combinatorial screening. The variables found to relate to the Y variable will be related through statistical models. Models to be considered fall into the generalized linear models framework using generalized least squares and regression tree models and others as applicable.

Step 3 — Model Diagnostics. Model diagnostics will include assessing the fit of the model and testing the assumptions implicit to the model. Of particular interest will be the spatial nature of the data and assumption of spatial independent or, alternatively, accounting for spatial dependence within the models.

Step 4 — Model Acceptance/Adjustment/Refinement. Results from the diagnostics will be used to either confirm the appropriateness of the model or influence adjustments or refinements to the model. Adjustments or refinements will require returning to step 2, model selection and fitting.

Step 5 — Model Output. Once an appropriate model is accepted, model output will include the model formula itself as well as metadata describing the user responsible for the model selection and the data used to drive the model.

2.3.2.4 Post-processing

The post-processing subsystem uses outputs from the Modeling Subsystem to generate visual and graphical products that are then made available to the user. In a typical scenario, the results of the OLS regression are applied to the input satellite and DEM data to provide a preliminary map of the total plants in the region. Kriged estimates of the residuals of this regression are then added to this map to produce an improved map. The postprocessing subsystem will typically produce a standard, application-interchangeable output file, such as an IDL file with ENVI header information, that can be used by other applications to reproject the data and overlay with other data layers as requested by the user. The final data products will be packaged with the appropriate metadata, assigned a unique data set identifier, and archived.

2.3.3 Backend Layer

The backend layer provides persistent storage for both system and user needs.

2.3.3.1 Archive

The ISFS backend consists of an archive subsystem logically comprising system, private, and public partitions. Subsystem control will be coordinated by a database that will store pointers to archived files. Initially, the files that we maintain will be contained in a logically arranged directory structure and indexed with a unique file ID. For externally stored data, the archive system will store a file ID and pointer or URL that can be used to retrieve and stage the archived files for subsequent processing.

3 ISFS Physical System Architecture

3.1 Hardware

Figure 5 illustrates the physical architecture of the ISFS. The User Interface Subsystem has client- and server-side components. Client workstations can be any computer equipped with a World Wide Web Consortium (W3C) standards compliant web browser. Some commonly available plug-ins may be required for certain functions, but the interface will push most of the complex rendering functions to the server. Our minimum supported display resolution will be 1024x768 pixels. The web server will be a standard two- or four processor Intel Pentium- or AMD Athlon-based computer running the GNU Linux operating system. The Apache http daemon will be running as a service using Secure Sockets Layer (SSL) on a TCP port that has agreed upon with the system administrator. We also will use a standard two- or four-processor Intel Pentium- or AMD Athlon-based computer running the GNU Linux operating system to host the Pre- and Postprocessing Subsystems. A separate similar system optimized for PostgreSQL will host the Ingest and Archive Subsystems. The preprocessing host will frontend the Modeling Subsystem but parallel code supporting core modeling operations, such as kriging, will be executed on various Beowulf clusters, including experimental clusters provided by the ESTO/CT program and, ultimately, a dedicated cluster that the project will build at the USGS site. Additional details about the cluster and all other aspects of the logical and physical design will be provided in subsequent versions of these software engineering documents.

3.2 Communications

The network transport between the client workstations and the Web Server will use Hyper Text Transfer Protocol (HTTP). For the user to upload new datasets a secure version of the File Transfer Protocol (FTP) will also be required. Both these protocols use TCP/IP (the Internet protocol) as the neutral layer for communication with the web server. We are considering two options for the network transport layer on the cluster. Gigabit Ethernet is a small fraction of the cost of the faster Myrinet and it is probably fast enough for our purposes. The cost of the increased I/O time must be balanced against the hardware cost and the complexities of implementing Myrinet. We will be performing engineering tradeoff studies to determine the final cluster configuration.

3.3 Implementation

The ISFS will be implemented using the best mix of scripting and programming languages to meet user and system development needs, including Apache, CHI, ION, Python, Java, and Postgres Client. The general approach to workflow control and process scheduling will be implemented through user/machine interactions that construct parameterized, bundled SQL and OS commands that can be queued, scheduled, and ultimately executed on the database host and Beowulf clusters. All activities will be bounded by execution status reports and inform the user of task readiness and run completeness and insure the integrity of the intermediary processing steps. The current system architecture assumes that all computers except for the client workstations will be PostgreSQL database clients. The database serves to coordinate application functions and manages the daemon processes running on the Web Server, Processing Hosts, Archive Host, and the Database Server itself. In addition the database will be the repository for required information to perform operations. The database class diagram in Figure 6 identifies the different types of data needed to support the ISFS system.

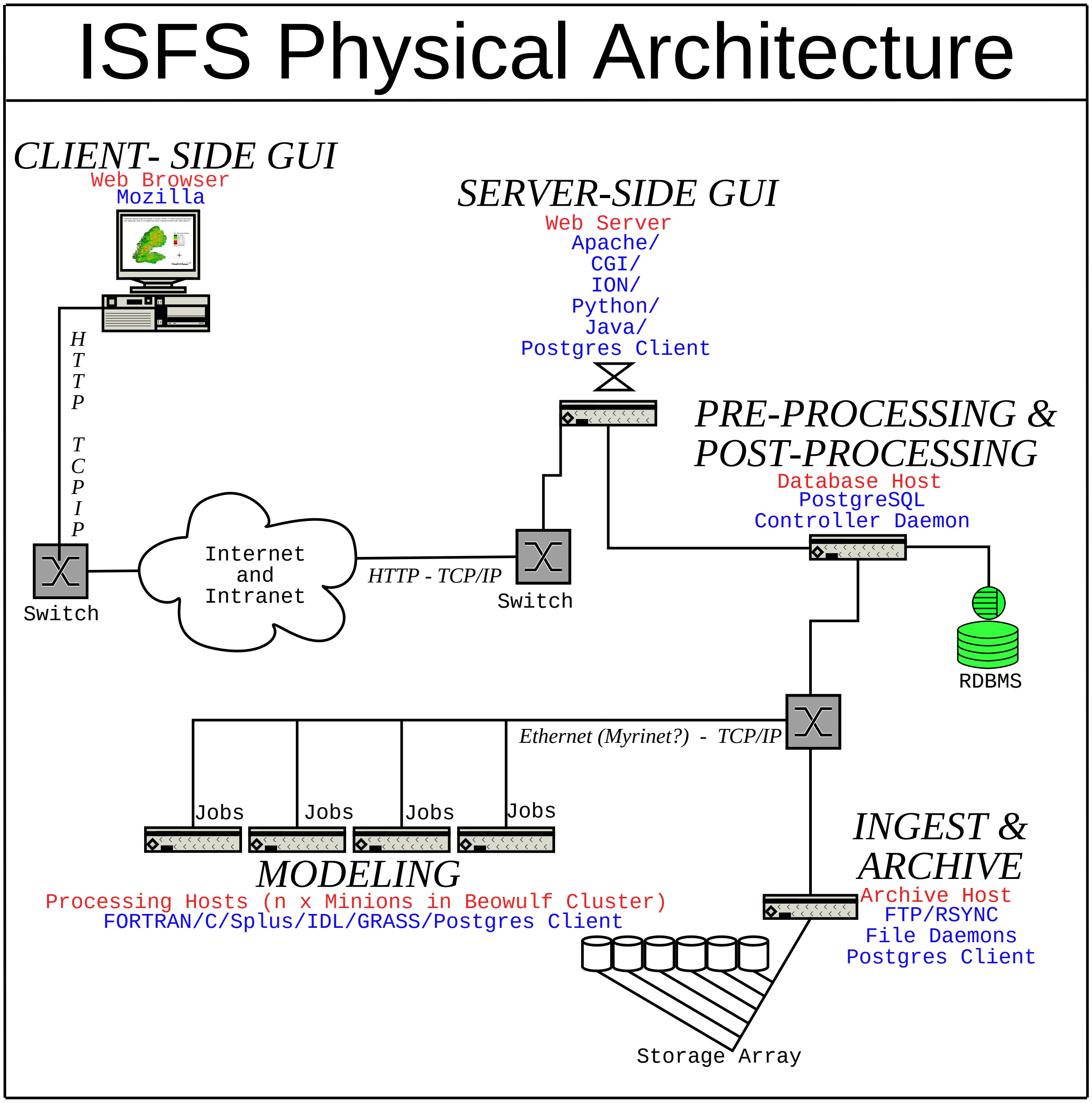


Figure 5. Physical System Architecture

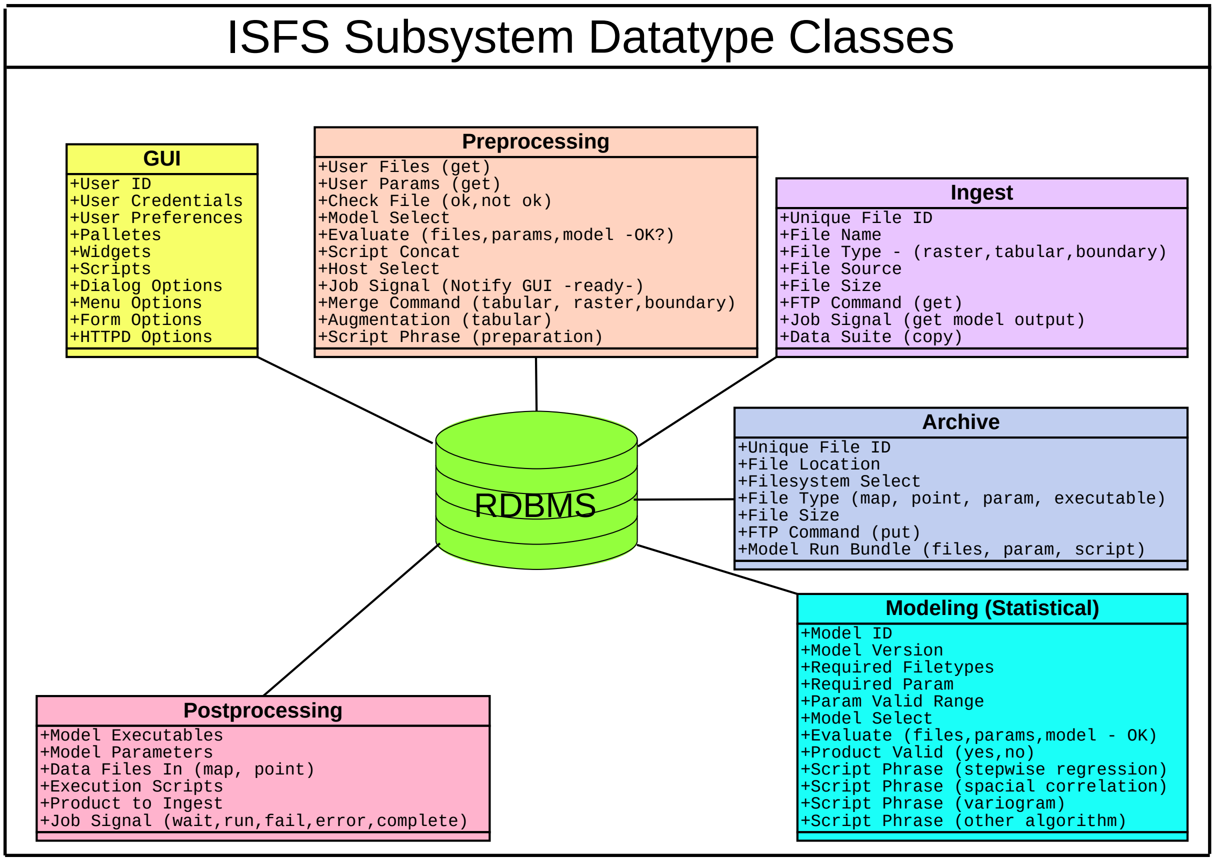


Figure 6. ISFS Database Classes

4 Use Cases

We have used the "Use Case" Model [1] to describe the system and subsystems from the point of view of various actors interacting with the system.

4.1 Actors

The primary actors and their general goals are:

Administrator has complete access to the system, monitors performance, and coordinates and authorizes system design changes, etc.

Model Builder is able to build tailored datasets as well as new analytical routines.

Model User is able to build tailored datasets to be used by existing analytical routines.

[1] Cockburn, Alistair, *Writing Effective Use Cases*, Addison-Wesley Pub Co, 2000

4.2 Use Case 1: Run a pre-defined model

Primary Actor: Model User

Goal: User runs a pre-defined model.

Scope: ISFS

Level: User Goal

Preconditions:

1. User account exists on the system.
2. Input files are accessible by the system.
3. Desired analysis routine integrated into the system.

Trigger:

Success End Condition: Model User has run the model and generated appropriate results.

Failed End Condition: Model User unable to obtain results.

Main Success Scenario:

1. Model User logs in to the system and starts a session.
2. Model User selects input files from list of available options, selects desired analysis routine from available options, and hits “Run” button.
3. Model User receives the output predictive map and uncertainty map.
4. Model User receives the associated metadata describing output files, run parameters, and performance stats.
5. Model User can optionally save the run results with personal annotations in personal repository.

Extensions:

Open Issues:

4.3 Use Case 2: Create a new data file for analysis

Primary Actor: Model Builder

Goal: Model Builder creates new data file and documentation for subsequent analysis by the modeling subroutine.

Scope: ISFS / Pre-Processing Subsystem

Level: User Goal

Preconditions:

1. User account exists on the system.
2. Input files in a format understood by the system.

Trigger: A new file to be analyzed is available.

Success End Condition: User has created new data file and documentation for subsequent analysis.

Failed End Condition: User unable to create useable data file.

Main Success Scenario:

1. Model Builder logs into the system and starts a session.
2. Model Builder selects to build a data suite.
3. Model Builder uploads a tabular data set.
4. Model Builder is informed that the data set uploaded into the system and can view the data set.
5. Model Builder views the data set.
6. Model Builder selects corresponding raster layers.
7. Model Builder activates the merging of raster variables into the tabular data.
8. Model Builder generates metadata documentation for the new data file.
9. Model Builder can optionally save the new data file with personal annotations in personal repository.

Extensions:

Open Issues:

4.4 Use Case 3: Create new explanatory variables

Primary Actor: Model User

Goal: Model User creates new variables to be used within the modeling subsystem.

Scope: ISFS / Modeling subsystem

Level: User Goal

Preconditions:

1. User account exists on the system.
2. Tabular Data Set: (multiple) raster data sets have been selected the user and ingested/merged with the system (see Use Case 2).

Trigger: User wishes to create additional variables beyond the “bands” contained in the raster data.

Success End Condition: Model Builder has created new variables for use in the modeling subsystem.

Failed End Condition: Model Builder unable to create new variables.

Main Success Scenario:

1. Model User logs in to the system and starts a session.
2. Model User navigates to the Modeling Subsystem and interface.
3. Model User specifies the response variable.
4. Model User selects variables to define spatial location.
5. Model User defines function of current layers used to create new layer/variable.
6. Model User uses the modeling subsystem to determine significant variables for prediction of the response variable.
7. Model User decides whether to include spatial model of residuals in the final prediction.
8. Model User chooses to view a map.
9. Model User is presented with image of model.

Extensions:

Open Issues:

4.5 Use Case 4: Create a new model for analysis

Primary Actor: Model Builder

Goal: Model Builder builds a new model.

Scope: ISFS / Pre-Processing Subsystem

Level: User Goal

Preconditions:

1. User account exists on the system.
2. Input files in a format understood by the system.

Trigger: A new file to be analyzed is available.

Success End Condition: User has created new model and metadata for subsequent analysis.

Failed End Condition: User unable to create useable model.

Main Success Scenario:

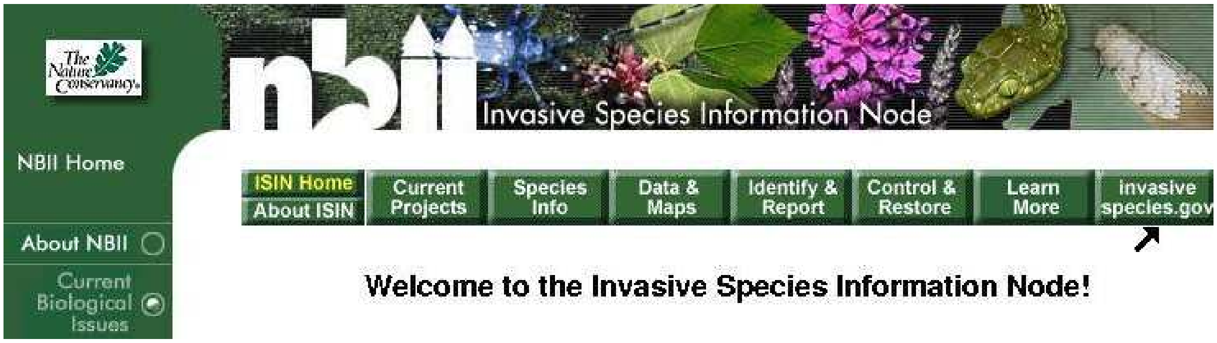
1. Model Builder logs into the system and starts a session.
2. Model Builder chooses to select a data suite.
3. Model Builder selects a data suite.
4. Model Builder chooses to view data suite.
5. Model Builder chooses to view existing model.
6. Model Builder chooses to create a new model.
7. Model Builder selects input files from list, selects variables to be included in analysis, and hits a “Run Data” button.
8. Model User receives the output predictive map and uncertainty map.
9. Model User receives the associated metadata describing output files, run parameters, and performance stats.
10. Model Builder can optionally save the results from the new model with annotations and associate the model output with the selected data suite.

Extensions:

Open Issues:

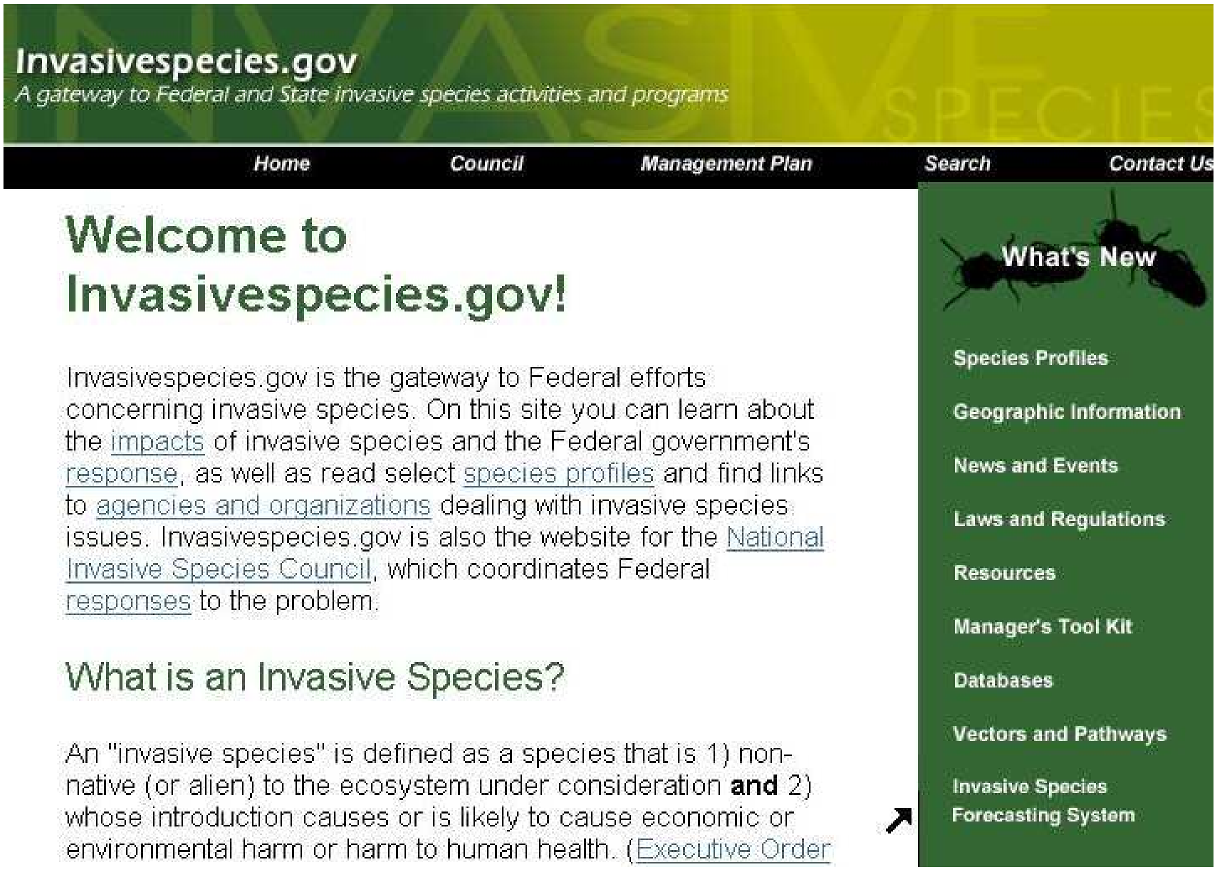
5 Use Case Illustrations

The system will have access from the various National Biological Information Infrastructure (NBII) member pages, as follows:



Welcome to the Invasive Species Information Node!

Figure 7. A link from the NBII Invasive Species Information Node.



Welcome to Invasivespecies.gov!

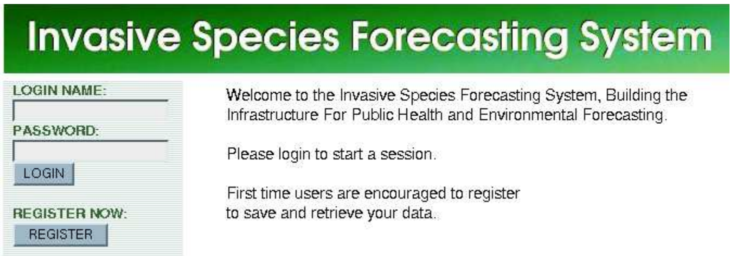
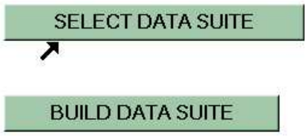
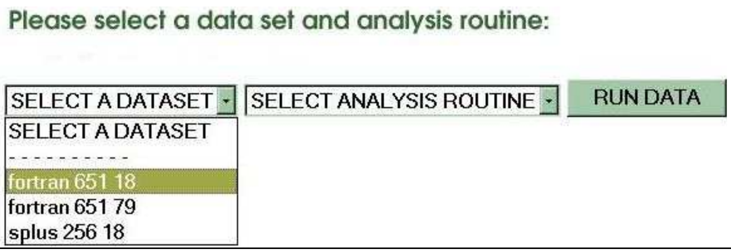
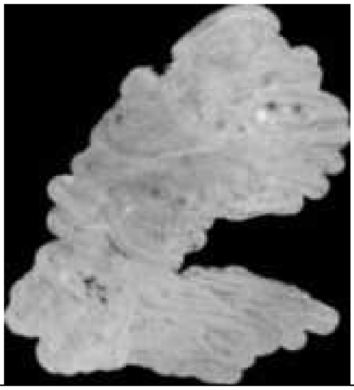
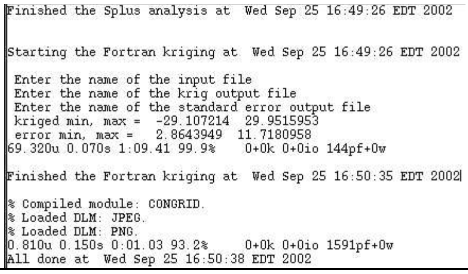
Invasivespecies.gov is the gateway to Federal efforts concerning invasive species. On this site you can learn about the impacts of invasive species and the Federal government's response, as well as read select species profiles and find links to agencies and organizations dealing with invasive species issues. Invasivespecies.gov is also the website for the National Invasive Species Council, which coordinates Federal responses to the problem.

What is an Invasive Species?

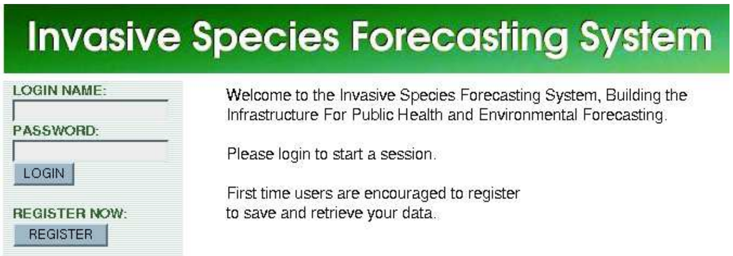
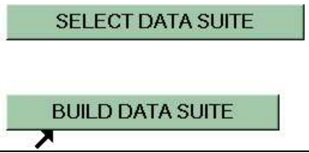
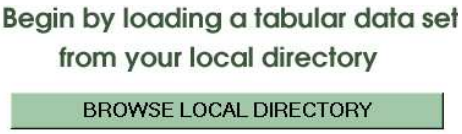
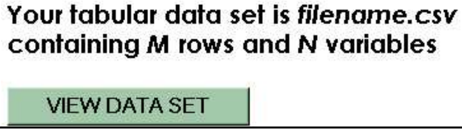
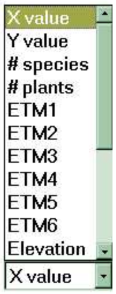
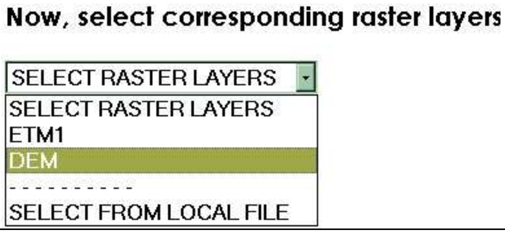
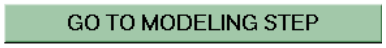
An "invasive species" is defined as a species that is 1) non-native (or alien) to the ecosystem under consideration **and** 2) whose introduction causes or is likely to cause economic or environmental harm or harm to human health. (Executive Order

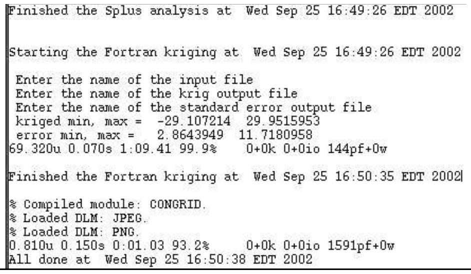
Figure 8. A link to the ISFS from the Invasivespecies.gov menu bar.

5.1 Use Case 1 Illustrated: Run a pre-defined model

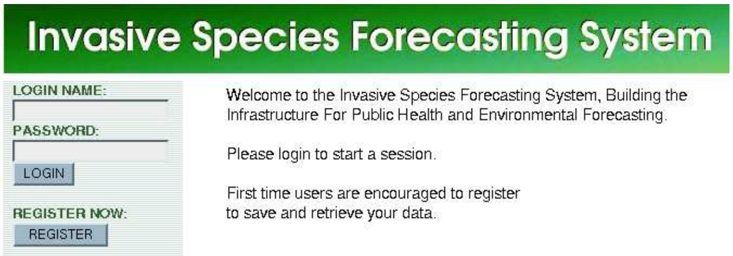
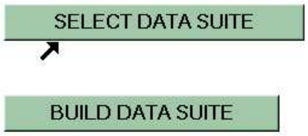
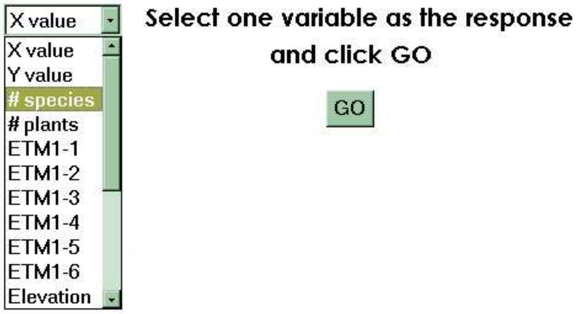
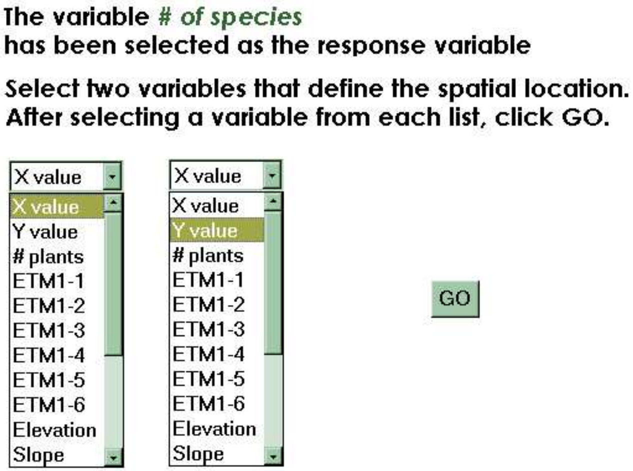
1. Model User logs into the system and starts a session	 Invasive Species Forecasting System Welcome to the Invasive Species Forecasting System, Building the Infrastructure For Public Health and Environmental Forecasting. Please login to start a session. First time users are encouraged to register to save and retrieve your data. LOGIN NAME: PASSWORD: LOGIN REGISTER NOW: REGISTER
2. Model User selects a data suite.	
3. Model User selects input files from list, selects desired analysis routine from available options, and hits a "Run Data" button.	 Please select a data set and analysis routine: SELECT A DATASET SELECT ANALYSIS ROUTINE RUN DATA SELECT A DATASET - - - - - - - - - - fortran 651 18 fortran 651 79 splus 256 18
4. Model User receives the output predictive map and uncertainty map.	
5. Model User receives the associated metadata describing output files, run parameters, and performance stats.	 Finished the Splus analysis at Wed Sep 25 16:49:26 EDT 2002 Starting the Fortran kriging at Wed Sep 25 16:49:26 EDT 2002 Enter the name of the input file Enter the name of the krig output file Enter the name of the standard error output file kriged min, max = -29.107214 29.9515953 error min, max = 2.8643949 11.7180958 69.320u 0.070s 1:09.41 99.9% 0+0k 0+0io 144pf+0w Finished the Fortran kriging at Wed Sep 25 16:50:35 EDT 2002 % Compiled module: CONGRID. % Loaded DLM: JPEG. % Loaded DLM: PNG. 0.810u 0.150s 0:01.03 93.2% 0+0k 0+0io 1591pf+0w All done at Wed Sep 25 16:50:38 EDT 2002
6. Model User can optionally save the run results with personal annotations in personal repository.	

5.2 Use Case 2 Illustrated: Create a new data file for analysis

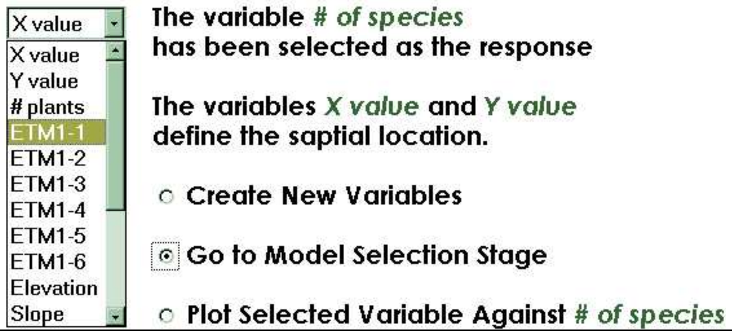
1. Model Builder logs into the system and starts a session	 Invasive Species Forecasting System LOGIN NAME: PASSWORD: LOGIN REGISTER NOW: REGISTER Welcome to the Invasive Species Forecasting System, Building the Infrastructure For Public Health and Environmental Forecasting. Please login to start a session. First time users are encouraged to register to save and retrieve your data.
2. Model Builder selects to build a data suite.	
3. Model Builder uploads a tabular data set.	 Begin by loading a tabular data set from your local directory
4. Model Builder is informed that the data set uploaded into the system and can view the data set.	 Your tabular data set is *filename.csv* containing *M* rows and *N* variables
5. Model Builder views the data set.	 X value Y value # species # plants ETM1 ETM2 ETM3 ETM4 ETM5 ETM6 Elevation X value
6. Model Builder selects corresponding raster layers.	 Now, select corresponding raster layers SELECT RASTER LAYERS SELECT RASTER LAYERS ETM1 DEM - - - - - - - - - - SELECT FROM LOCAL FILE
7. Model Builder activates the merging of raster variables into the tabular data.	

8. Model Builder generates metadata documentation for the new data file.	 Finished the Splus analysis at Wed Sep 25 16:49:26 EDT 2002 Starting the Fortran kriging at Wed Sep 25 16:49:26 EDT 2002 Enter the name of the input file Enter the name of the krig output file Enter the name of the standard error output file kriged min, max = -29.107214 29.9515953 error min, max = 2.8643949 11.7180958 69.320u 0.070s 1:09.41 99.9% 0+0k 0+0io 144pf+0w Finished the Fortran kriging at Wed Sep 25 16:50:35 EDT 2002 % Compiled module: CONGRID. % Loaded DLM: JPEG. % Loaded DLM: PNG. 0.810u 0.150s 0:01.03 93.2% 0+0k 0+0io 1591pf+0w All done at Wed Sep 25 16:50:38 EDT 2002
9. Model Builder can optionally save the new data file with personal annotations in personal repository.	

5.3 Use Case 3 Illustrated: Create new explanatory variable(s)

1. Model User logs into the system and starts a session.	 Invasive Species Forecasting System LOGIN NAME: PASSWORD: LOGIN REGISTER NOW: REGISTER Welcome to the Invasive Species Forecasting System, Building the Infrastructure For Public Health and Environmental Forecasting. Please login to start a session. First time users are encouraged to register to save and retrieve your data.
2. Model User navigates to the Modeling Subsystem and interface.	
3. Model User specifies the response variable.	 Select one variable as the response and click GO X value X value Y value # species # plants ETM1-1 ETM1-2 ETM1-3 ETM1-4 ETM1-5 ETM1-6 Elevation GO
4. Model User selects variables to define spatial location.	 The variable *# of species* has been selected as the response variable Select two variables that define the spatial location. After selecting a variable from each list, click GO. X value X value Y value # plants ETM1-1 ETM1-2 ETM1-3 ETM1-4 ETM1-5 ETM1-6 Elevation Slope X value X value Y value # plants ETM1-1 ETM1-2 ETM1-3 ETM1-4 ETM1-5 ETM1-6 Elevation Slope GO

<table>
<tr>
<td>5. Model User defines function of current layers used to create new layer/variable.</td>
<td>



The variable # of species
has been selected as the response

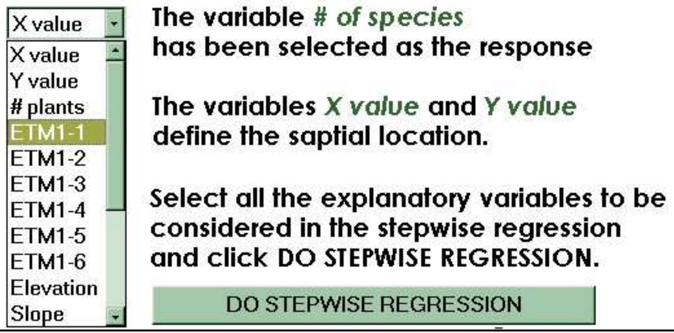
The variables X value and Y value
define the saptial location.

○ Create New Variables

● Go to Model Selection Stage

○ Plot Selected Variable Against # of species

</td>
</tr>
<tr>
<td>6. Model User uses the modeling subsystem to determine significant variables for prediction of the response variable.</td>
<td>

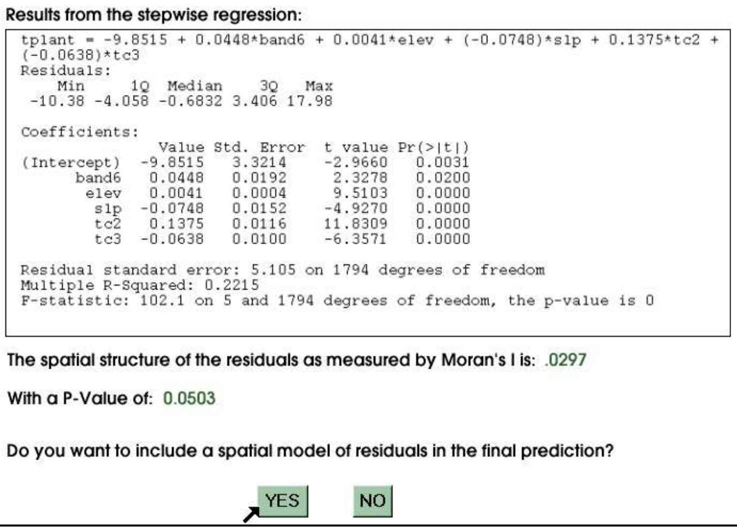


The variable # of species
has been selected as the response

The variables X value and Y value
define the saptial location.

Select all the explanatory variables to be
considered in the stepwise regression
and click DO STEPWISE REGRESSION.

</td>
</tr>
<tr>
<td>7. Model User decides whether to include spatial model of residuals in the final prediction.</td>
<td>



Results from the stepwise regression:

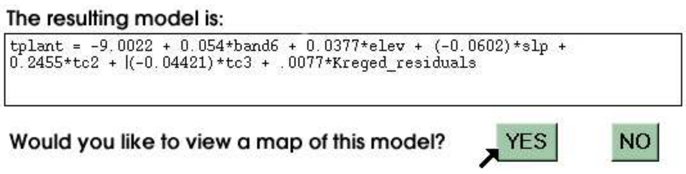
```
tplant = -9.8515 + 0.0448*band6 + 0.0041*elev + (-0.0748)*slp + 0.1375*tc2 +
(-0.0638)*tc3
Residuals:
    Min      1Q  Median     3Q   Max
 -10.38 -4.058 -0.6832 3.406 17.98

Coefficients:
              Value Std. Error  t value Pr(>|t|)
(Intercept)  -9.8515   3.3214    -2.9660   0.0031
      band6   0.0448   0.0192     2.3278   0.0200
       elev   0.0041   0.0004     9.5103   0.0000
        slp  -0.0748   0.0152    -4.9270   0.0000
        tc2   0.1375   0.0116    11.8309   0.0000
        tc3  -0.0638   0.0100    -6.3571   0.0000

Residual standard error: 5.105 on 1794 degrees of freedom
Multiple R-Squared: 0.2215
F-statistic: 102.1 on 5 and 1794 degrees of freedom, the p-value is 0
```

The spatial structure of the residuals as measured by Moran's I is: .0297

With a P-Value of: 0.0503

Do you want to include a spatial model of residuals in the final prediction?

</td>
</tr>
<tr>
<td>8. Model User chooses to view a map of model.</td>
<td>

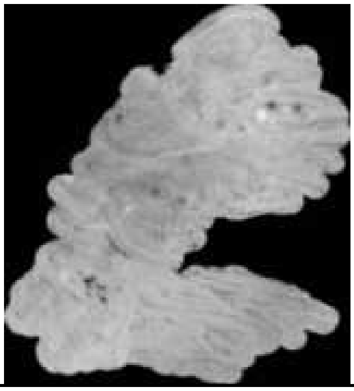


The resulting model is:

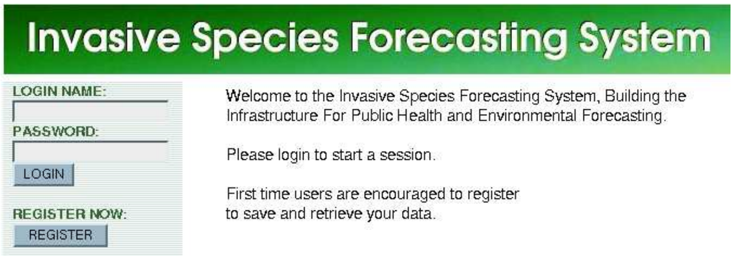
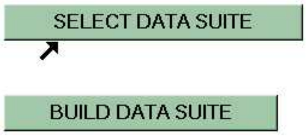
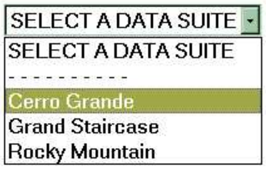
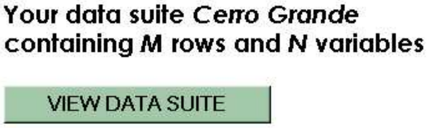
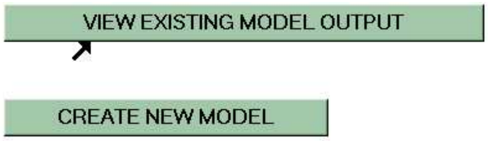
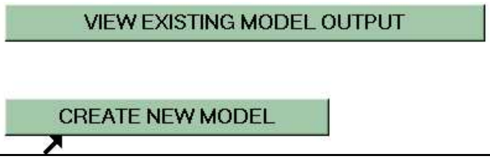
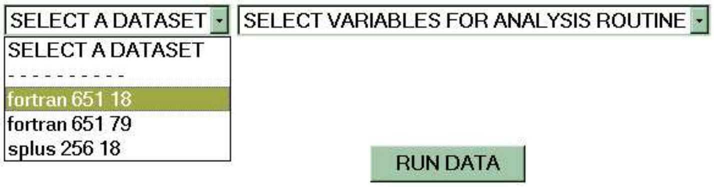
```
tplant = -9.0022 + 0.054*band6 + 0.0377*elev + (-0.0602)*slp +
0.2455*tc2 + |(-0.04421)*tc3 + .0077*Kreged_residuals
```

Would you like to view a map of this model?

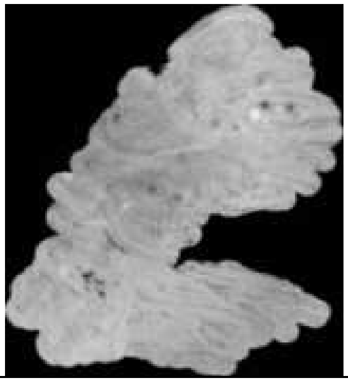
</td>
</tr>
</table>

9. Model User is presented with image of model.	

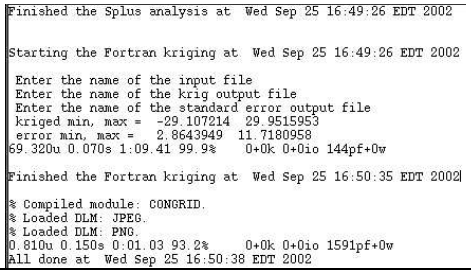
5.4 Use Case 4 Illustrated: Create a new model for analysis

1. Model Builder logs into the system and starts a session.	 Invasive Species Forecasting System LOGIN NAME: PASSWORD: LOGIN REGISTER NOW: REGISTER Welcome to the Invasive Species Forecasting System, Building the Infrastructure For Public Health and Environmental Forecasting. Please login to start a session. First time users are encouraged to register to save and retrieve your data.
2. Model User chooses to select a data suite.	
3. Model Builder selects a data suite.	 SELECT A DATA SUITE SELECT A DATA SUITE - - - - - - - - - - Cerro Grande Grand Staircase Rocky Mountain
4. Model Builder chooses to view data suite.	Your data suite *Cerro Grande* containing *M* rows and *N* variables
5. Model Builder chooses to view existing model.	
6. Model Builder chooses to create a new model.	
7. Model User selects input files from list, selects variables to be included in analysis, and hits a "Run Data" button.	 SELECT A DATASET SELECT VARIABLES FOR ANALYSIS ROUTINE SELECT A DATASET - - - - - - - - - - fortran 651 18 fortran 651 79 splus 256 18 RUN DATA

<table>
<tr>
<td>8. Model User receives the output predictive map and uncertainty map.</td>
<td>



</td>
</tr>
<tr>
<td>9. Model User receives the associated metadata describing output files, run parameters, and performance stats.</td>
<td>



```
Finished the Splus analysis at  Wed Sep 25 16:49:26 EDT 2002


Starting the Fortran kriging at  Wed Sep 25 16:49:26 EDT 2002

 Enter the name of the input file
 Enter the name of the krig output file
 Enter the name of the standard error output file
 kriged min, max =  -29.107214  29.9515953
 error min, max =   2.8643949  11.7180958
69.320u 0.070s 1:09.41 99.9%    0+0k 0+0io 144pf+0w

Finished the Fortran kriging at  Wed Sep 25 16:50:35 EDT 2002

% Compiled module: CONGRID.
% Loaded DLM: JPEG.
% Loaded DLM: PNG.
0.810u 0.150s 0:01.03 93.2%     0+0k 0+0io 1591pf+0w
All done at  Wed Sep 25 16:50:38 EDT 2002
```

</td>
</tr>
<tr>
<td>10. Model Builder can optionally save the results from the new model with annotations and associate the model output with the selected data suite.</td>
<td>

</td>
</tr>
</table>

A Glossary

BP Biotic Prediction project
CT Computational Technologies project
CONOP Concept of Operations
COTS Commercial Off The Shelf
CSU Colorado State University
DAAC Distributed Active Archive Center
ESTO Earth Science Technology Office
FTP File Transfer Protocol
GSFC Goddard Space Flight Center
GUI Graphical User Interface
HTTP Hyper-Text Transport Protocol
ISFS Invasive Species Forecasting System
NBII National Biological Information Infrastructure
NREL Natural Resources Ecology Laboratory
RDBMS Relational Database Management System
SEP Software Engineering / Development Plan
USGS United States Geological Survey